

FIG. 1 is a block diagram of a computer system 10. The computer system 10 includes an instruction processor (IP) 11, an operating system (OS) 12, an input/output module (IOM) 13, and a task control unit (TCU) 14. The instruction processor (IP) 11 includes an operating system (OS) 12, which includes a features list 23, supported features 24, and a report 25. The features list 23 includes features required/optional 24. The supported features 24 includes features required/optional 25. The report 25 is connected to the features list 23. The operating system (OS) 12 includes an exchange control 21, which is connected to the features list 23 and the supported features 24. The exchange control 21 is also connected to a TCU_exchange_features function 16. The TCU_exchange_features function 16 is connected to hardware support for function calls 14. The input/output module (IOM) 13 includes a task control unit (TCU) 14, which includes a features list 40, supported features 41, and a report 42. The features list 40 includes features required/optional 41. The supported features 41 includes features required/optional 42. The report 42 is connected to the features list 40. The task control unit (TCU) 14 includes an exchange control 21, which is connected to the features list 40 and the supported features 41. The exchange control 21 is also connected to a TCU_exchange_features function 16. The TCU_exchange_features function 16 is connected to hardware support for function calls 14.

FIG. 1

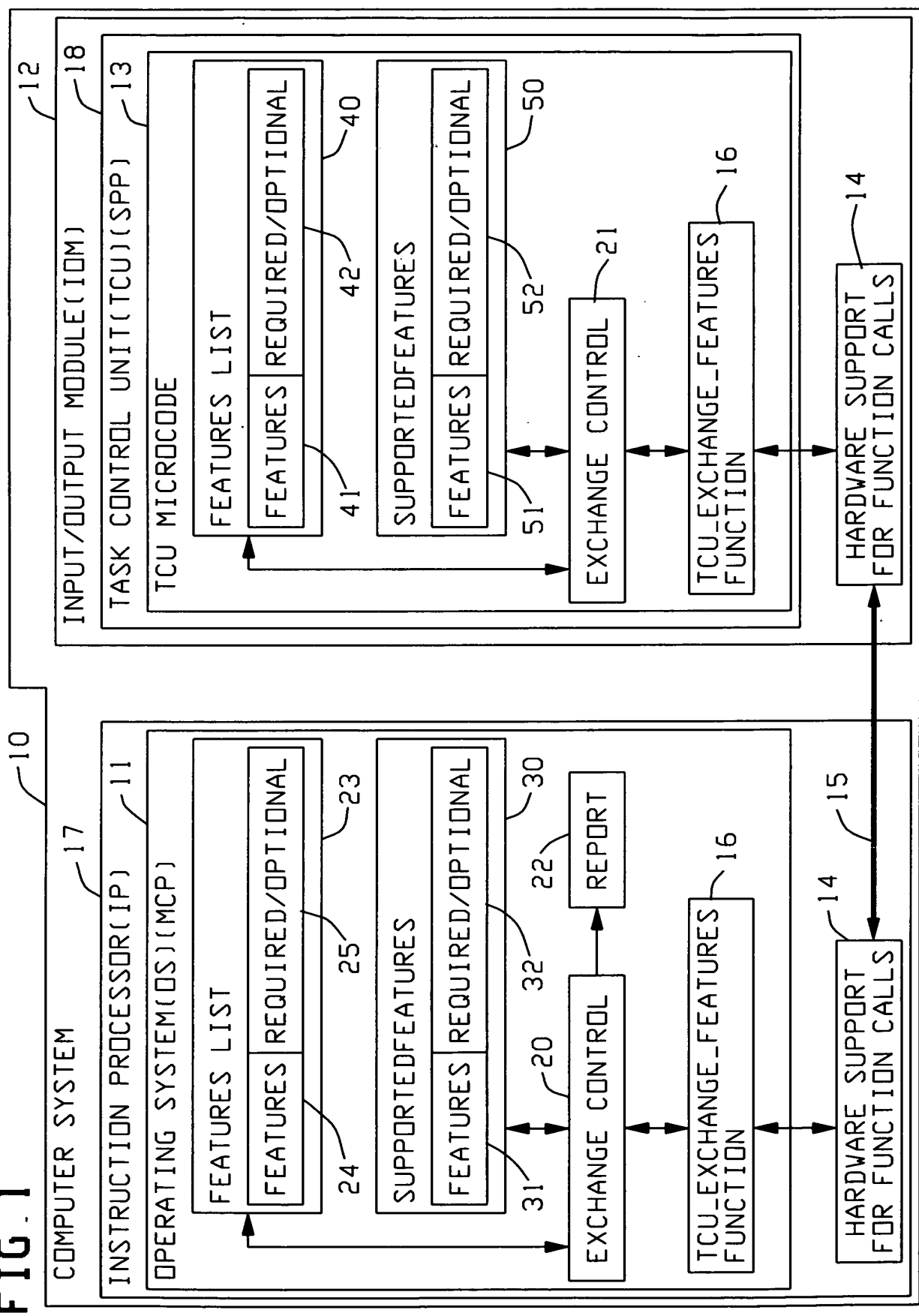


FIG. 2(a)

```

BOOLEAN PROCEDURE
TCU_EXCHANGE_FEATURES
(WORDNUM, MCPTCUFEATURES,
LASTCALL);
NAME      WORDNUM, MCPTCUFEATURES,
          LASTCALL;
INTEGER WORDNUM;
BOOLEAN MCPTCUFEATURES, LASTCALL;
  
```

FIG. 2(b)

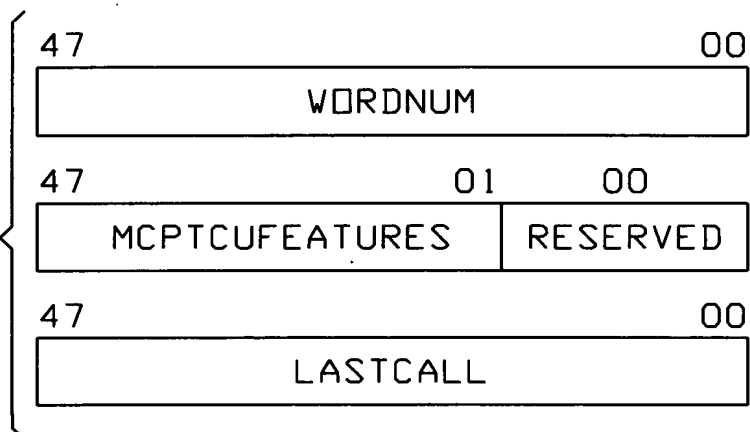


FIG. 2(c)

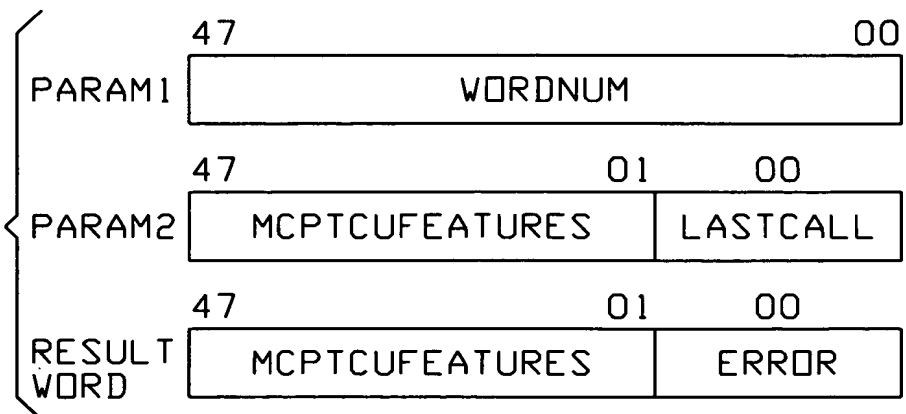
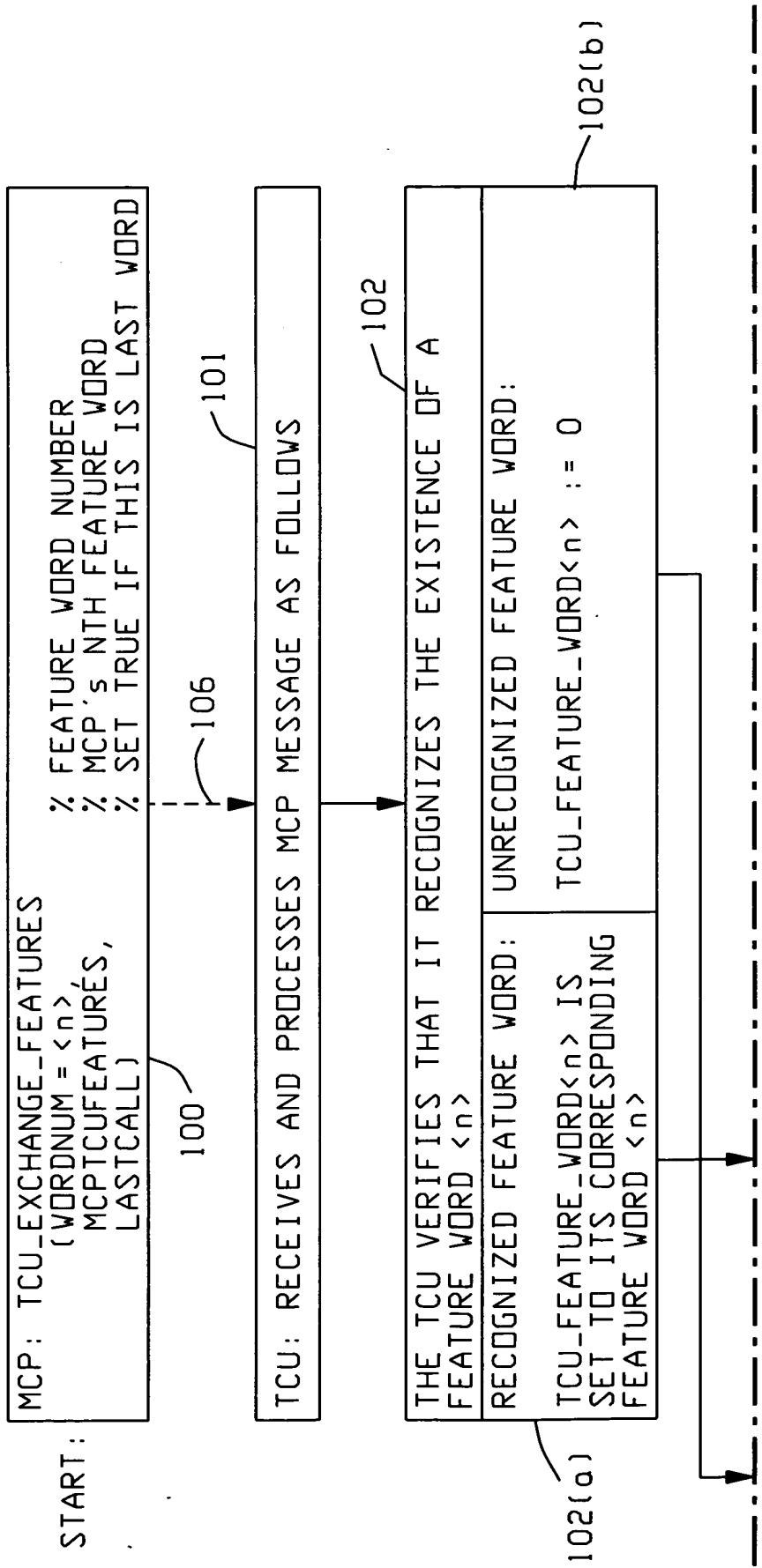


FIG. 3(a)A

FIG. 3(a)

FIG. 3(a)A
FIG. 3(a)B



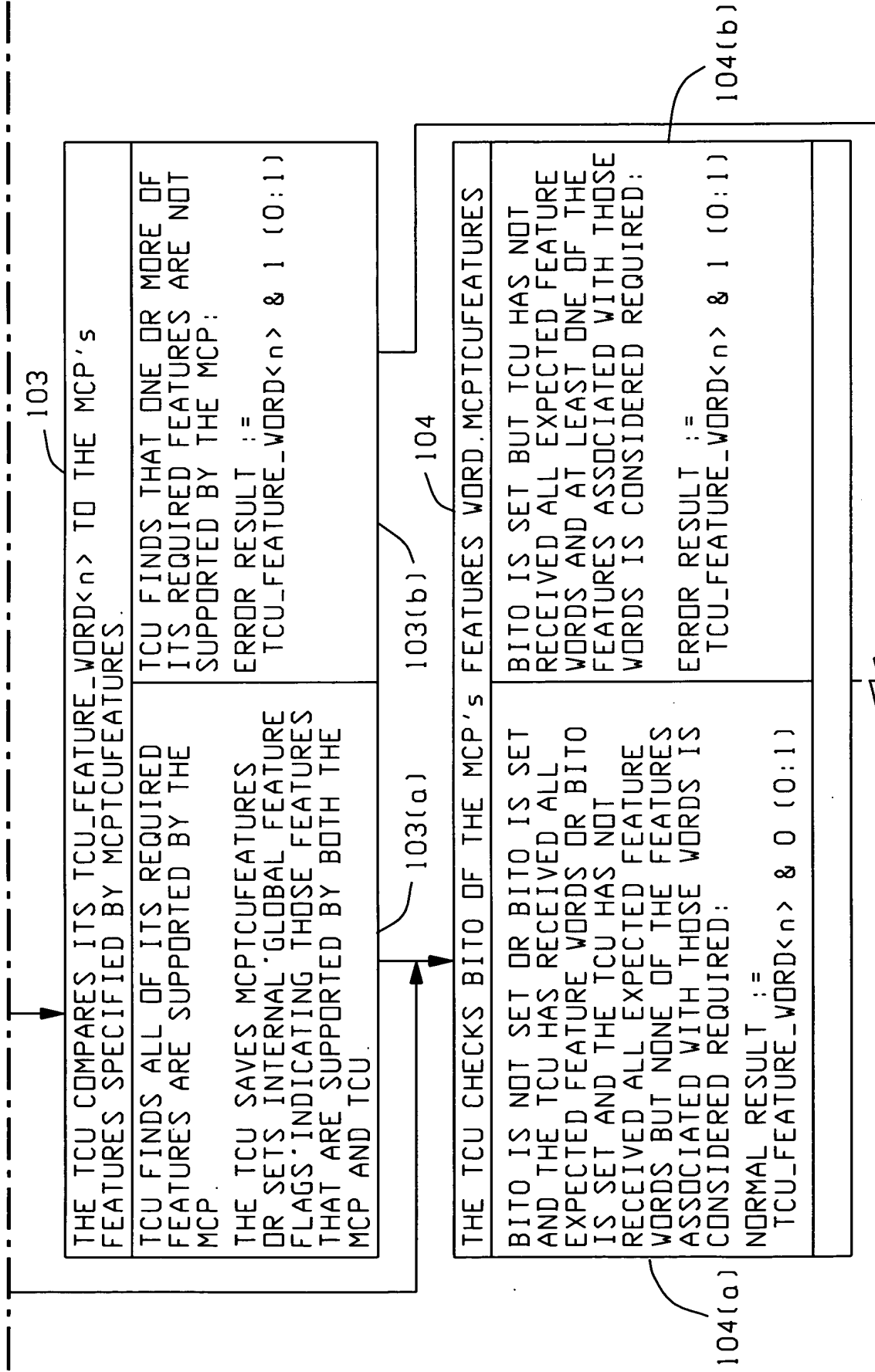
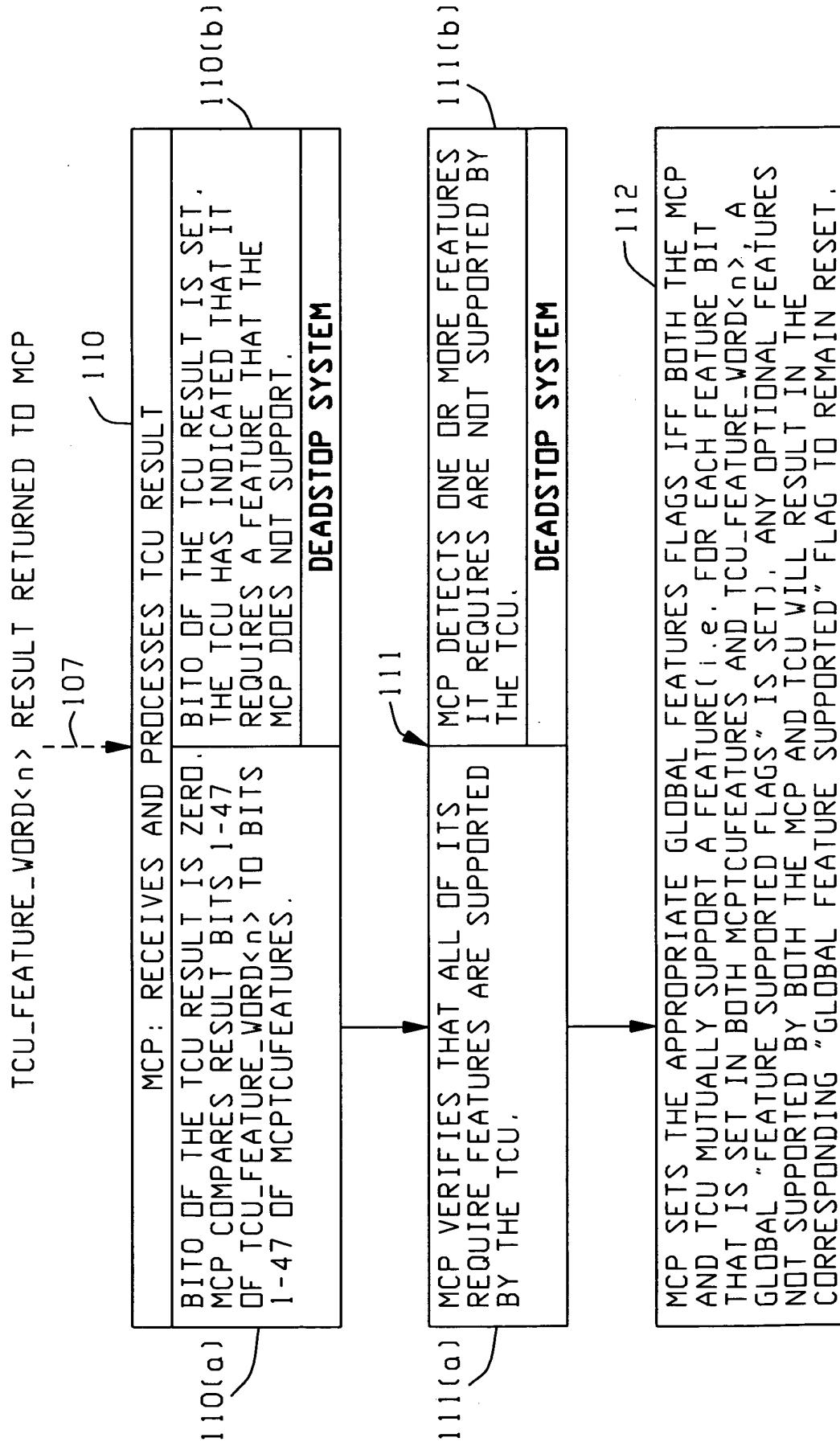


FIG. 3(a)B RETURN TCU_FEATURE_WORD<N> RESULT TO MCP

FIG. 3(b)A



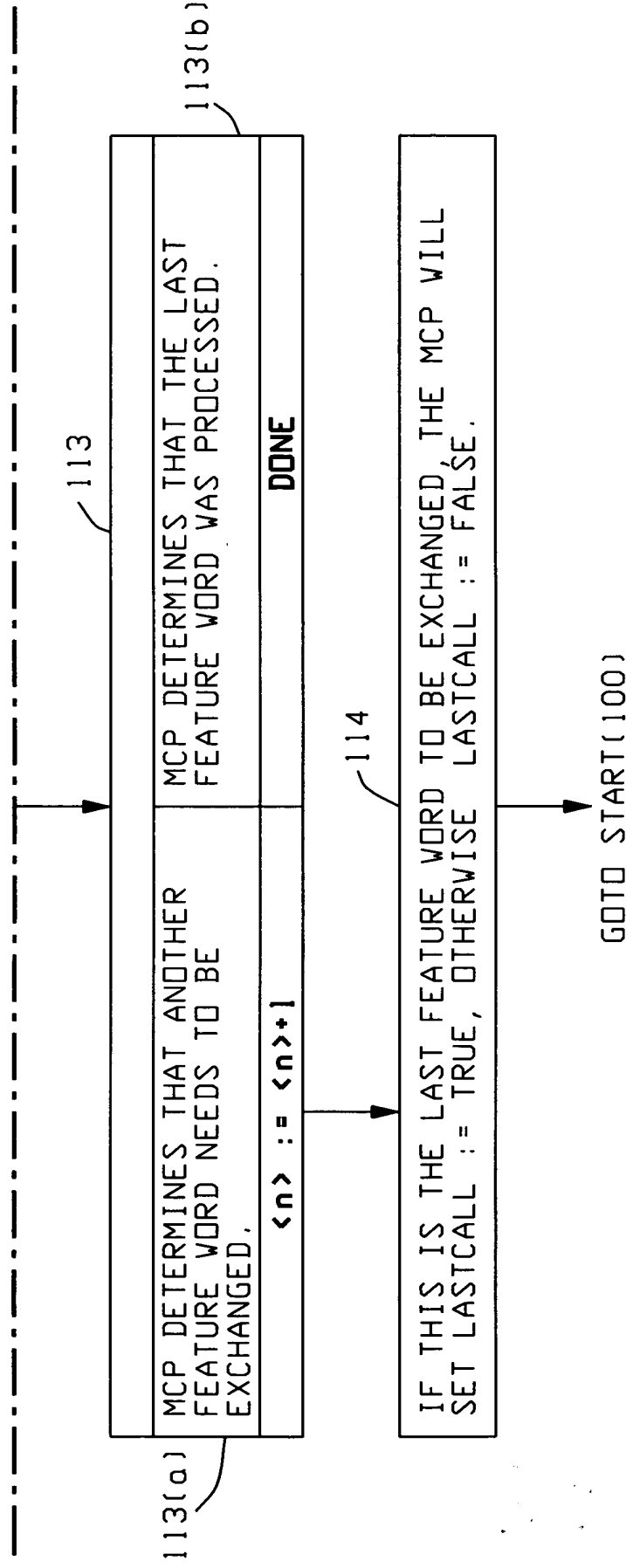


FIG. 3(b)B

FIG. 3(b)A
FIG. 3(b)B

FIG. 3(b)

```

START:  [OS]  WORDNUM_OS := 1;

LOOP:   FEATUREWORD_OS := FEATURES_OS[WORDNUM_OS];  % FEATURES_OS: Array of feature word
                                                    % bit masks supported by OS. This
                                                    % is hardcoded data.

        IF this is the last feature word THEN
            FEATUREWORD_OS := FEATUREWORD_OS & 1 [0:1];

        [OS]  SPPFEATURES_OS := EXCHANGE_FEATURES (WORDNUM_OS, FEATUREWORD_OS);

        [SPP] %Receives WORDNUM_OS and FEATUREWORD_OS from function call. Note that references
              % to these parameters use "<>"
        IF <WORDNUM_OS> is not a recognized feature word THEN
            FEATUREWORD_SPP := 0;
            GOTO CHECK_LAST;

        FEATUREWORD_SPP := FEATURES_SPP[<WORDNUM_OS>];  %FEATURES_SPP: Array of feature word
                                                    %bit masks supported by SPP. This
                                                    % is hardcoded data.

        IF (FEATUREWORD_SPP NEQ <FEATUREWORD_OS>) THEN % Compare bits [47:46]
            IF a feature required by SPP is not supported by OS THEN
                RESULT_SPP := FEATUREWORD_SPP & 1 [0:1]; % Set error bit in result
                GOTO RETURN;
            % SUPPORTEDFEATURES_SPP: Array of supported features bit masks.
            SUPPORTEDFEATURES_SPP[<WORDNUM_OS>] := FEATUREWORD_SPP AND <FEATUREWORD_OS>;
        ELSE
            SUPPORTEDFEATURES_SPP[<WORDNUM_OS>] := FEATUREWORD_SPP;

```

Fig. 4A

Fig. 4B

Figure 4

```

CHECK_LAST:      IF Bit0 of <FEATUREWORD_OS> set AND did not receive all expected feature words THEN
                  IF any of the remaining features are required by the SPP THEN
                    RESULT_SPP := FEATUREWORD_SPP & 1 [0:1]; % Set error bit in result
                    GOTO RETURN;
                  ELSE
                    % Set remaining words in SUPPORTEDFEATURES_SPP array to zero;
                  ELSE
                    RESULT_SPP := FEATUREWORD_SPP & 0 [0:1]; % Non-error Result (reset error bit)

RETURN:          RETURN (RESULT_SPP);

[OS]            IF bit0 of SPPFEATURES_OS is set THEN
                  %Fatal error. Abort system initialization. Report feature mismatch to
                  %operations, etc. System Stopped.

                  IF (FEATUREWORD_OS NEQ SPPFEATURES_OS) THEN
                    IF a feature required by OS is not supported by SPP THEN
                      % Fatal error. Abort system initialization. Report feature mismatch to
                      % operations, etc. System Stopped.
                    ELSE
                      % SUPPORTEDFEATURES_OS: Array of supported features bit masks.
                      SUPPORTEDFEATURES_OS[WORDNUM_OS] := FEATUREWORD_OS AND SPPFEATURES_OS;
                    ELSE
                      SUPPORTEDFEATURES_OS[WORDNUM_OS] := FEATUREWORD_OS;

                  IF more feature words to exchange THEN
                    BEGIN
                      WORDNUM_OS := WORDNUM_OS + 1;
                      GOTO LOOP;
                    END;

```

Fig. 4A
Fig. 4B

Figure 4